EXHIBIT R

Exhibit 10 – Belanovic and Leeser

’156 Patent

Claim Limitation (Claim 7)	Exemplary Disclosure
[**156a**] A device comprising:	As reflected in Pavle Belanovic’s thesis, *Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application*, and his related article with Miriam Leeser, *Library of Parameterized Floating-Point Modules and Their Use*, they made and used a system that consists of a device. *See, e.g.*: “**The Wildstar Reconfigurable Computing Engine** Reconfigurable computing is characterized by use of hardware elements that have reconfigurable architectures, as opposed to general purpose computing which uses hardware elements with fixed architectures. Many reconfigurable computing systems are based on one or more FPGAs connected to a number of memory banks. All designs presented in this project are implemented on the Wildstar reconfigurable computing engine from Annapolis Micro Systems. Figure 1.2 shows the structure of this board. Some of the main features of the Wildstar board are: • 3 Xilinx VIRTEX XCV1000 FPGAs, • total of 3 million system gates, • 40 Mbytes of SRAM, • 1.6 Gbytes/sec I/O bandwidth, • 6.4 Gbytes/sec memory bandwidth, • processing clock rates up to 100MHz.” Belanovic, *Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application* at 14. “For synthesis and mapping of all designs in this project we used Synplicity Pro from Synplify. Mapping, placing and routing of the designs was done using Xilinx Alliance tools. In order to verify the fidelity of the VHDL descriptions to the intended

Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 7)	Exemplary Disclosure
	functionality, all designs in this project were simulated with Mentor Graphics ModelSim prior to being implemented in hardware." Belanovic, *Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application* at 13-14.
[**156b**] at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,	As reflected in Pavle Belanovic's thesis, *Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application*, and his related article with Miriam Leeser, *Library of Parameterized Floating-Point Modules and Their Use*, they made and used a system with at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first input signal representing a first numerical value to produce a first output signal representing a second numerical value. *See, e.g.*: "Thus, module parameterized priority encoder has been developed, taking a signal to be examined on its input and producing the value, in unsigned fixed-point representation, of the index of the most significant '1' in the input signal. The module is parameterized by the width of the input signal, as well as the width of the output signal." Belanovic, *Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application* at 25-26. "The final stage of both the signed and the unsigned architectures is the output stage, where the computed fixed-point representation is placed on the output, unless the input was zero or an exception was encountered during operation or received at the input, in which case the output is set to all zeros." Belanovic, *Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application* at 43. *See also* Belanovic, *Library of Parameterized Hardware* at 15 (Fig 1.2) (showing 32-bit inputs and outputs to PE 1 and PE 2).

Exhibit 10 – Belanovic and Leeser

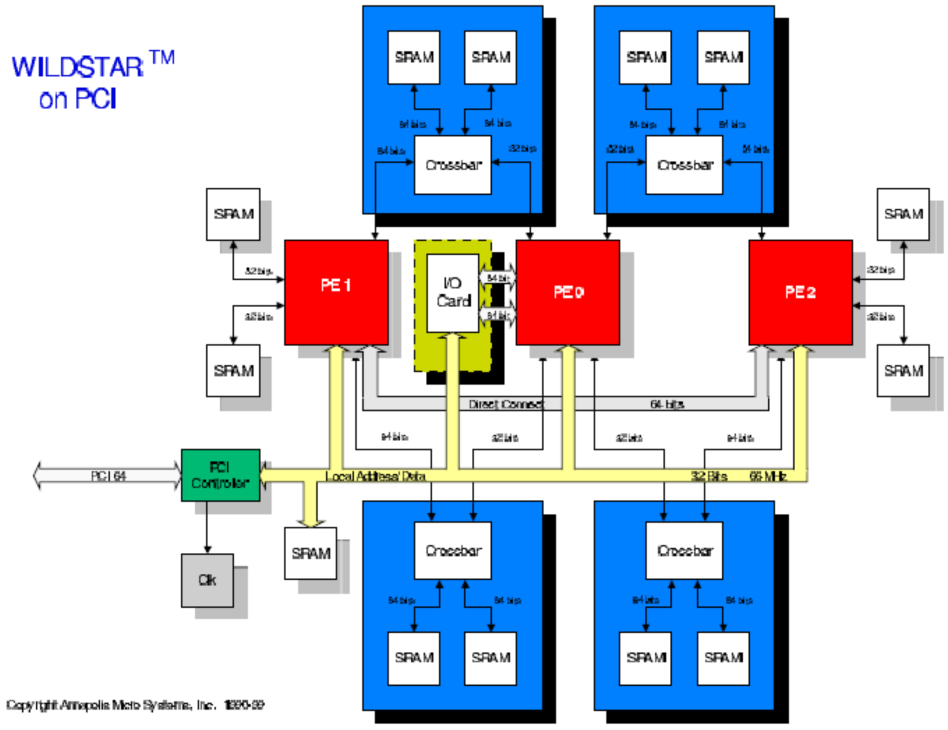
Claim Limitation (Claim 7)	Exemplary Disclosure
	 WILDSTAR TM on PCI SRAM Crossbar 64 bits 32 bits PE 1 PE 0 PE 2 I/O Card Direct Connect 64 bits PCI 64 PCI Controller Local Address/ Data 32 Bits 66 MHz Clk Copyright Annapolis Micro Systems, Inc. 1998-99 Figure 1.2: Structure of the Wildstar reconfigurable computing engine *See also* Belanovic, *Library of Parameterized Hardware* at 35 (Fig 2.4) (showing the inputs and outputs to the floating-point multiplier); *see generally* Belanovic, *Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application* at 35-43 (Section 2.4) (explaining hardware models for conversation).

Exhibit 10 – Belanovic and Leeser

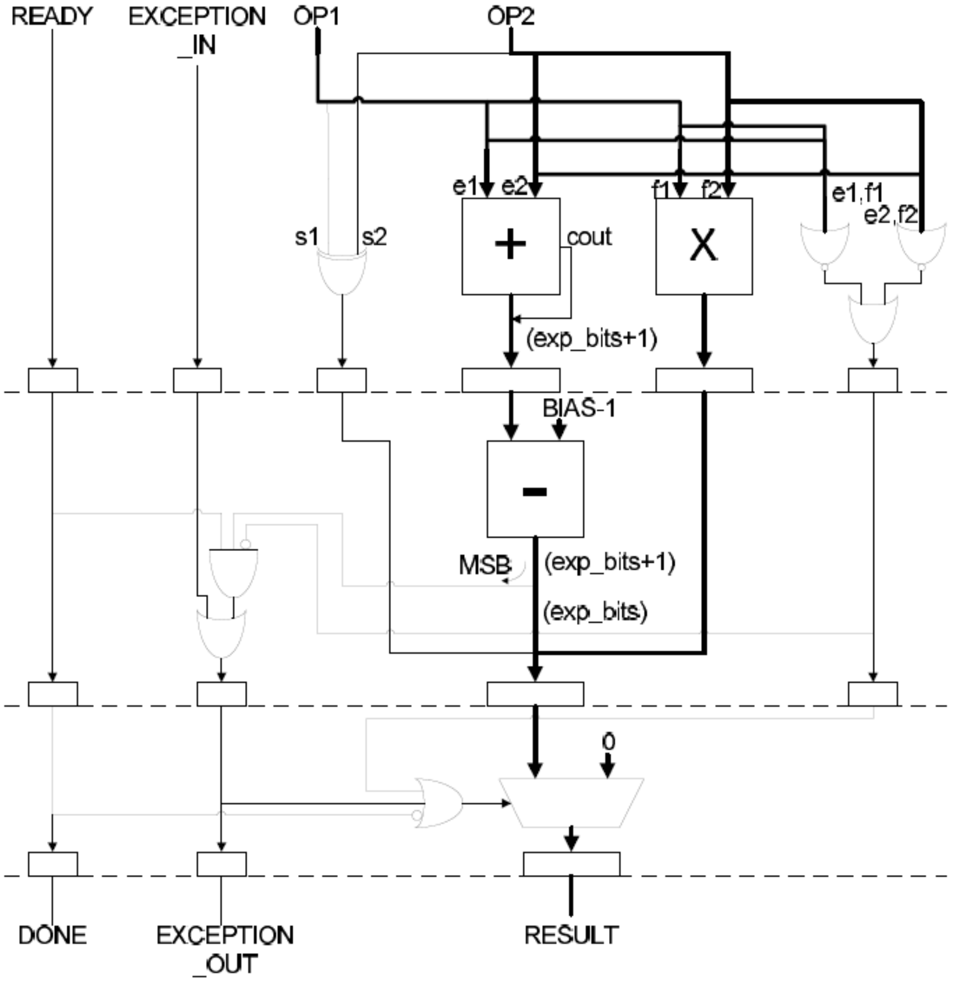
Claim Limitation (Claim 7)	Exemplary Disclosure
	 READY EXCEPTION _IN OP1 OP2 e1 e2 f1 f2 e1,f1 e2,f2 s1 s2 + cout X (exp_bits+1) BIAS-1 - MSB (exp_bits+1) (exp_bits) 0 DONE EXCEPTION _OUT RESULT Figure 2.4: Floating-point multiplication “We have developed a library of fully parameterized hardware modules for format

Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 7)	Exemplary Disclosure
	control, arithmetic operations and conversion to and from any fixed-point format. The format converters allow for hybrid implementations that combine both fixed and floating-point calculations. This permits the designer to choose between the increased range of floating-point and the increased precision of fixed-point within the same application." Belanovic and Leeser, *Library of Parameterized Floating-Point Modules and Their Use* at 1. "A natural tradeoff exists between smaller bitwidths requiring fewer hardware resources and higher bitwidths providing better precision. Also, within a given total bitwidth, it is possible to assign various combinations of bitwidths to the exponent and fraction fields, where wider exponents result in higher range and wider fractions result in better precision. . . . Often, much smaller bitwidths than those specified in the 754 standard are sufficient to provide the desired precision. Reduced bitwidth implementations require fewer resources and thus allow for more parallel implementations than using the full IEEE standard format. In custom hardware designs, it is possible, and indeed desirable, to have full control and flexibility over the exact floating-point format implemented." Belanovic and Leeser, *Library of Parameterized Floating-Point Modules and Their Use* at 2.
[**156c**] wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the	As reflected in Pavle Belanovic's thesis, *Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application*, and his related article with Miriam Leeser, *Library of Parameterized Floating-Point Modules and Their Use*, they made and used a system with the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input. *See, e.g.*: "The floating-point formats in our work are a generalized superset of all these formats.

Exhibit 10 – Belanovic and Leeser

<table>
<tr><th>Claim Limitation (Claim 7)</th><th>Exemplary Disclosure</th></tr>
<tr><td>LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input; and</td><td>It includes all the IEEE formats as particular instances of exponent and mantissa bitwidths, as well as the flexible floating-point format presented by Dido et al.[2] and the two formats by Shirazi et al.[17].” Belanovic, <i>Library of Parameterized Hardware</i> at 19.

“The experiments were conducted by synthesizing the modules for specific floating-point formats on the Annapolis Micro Systems Wildstar reconfigurable computing engine (see Section 1.3). Table 2.2 shows results of the synthesis experiments on floating-point operator modules. The quantities for the area of each instance are expressed in slices of the Xilinx XCV1000 FPGA. Results for the fp_add module in Table 2.2 also represent the fp_sub module, which has the same amount of logic.

Floating-point formats used in the experiments were chosen to represent the range of realistic floating-point formats from 8 to 32 bits in total bitwidth and include the IEEE single precision format (E1 in Table 2.2).” Belanovic, <i>Library of Parameterized Hardware</i> at 46-47.

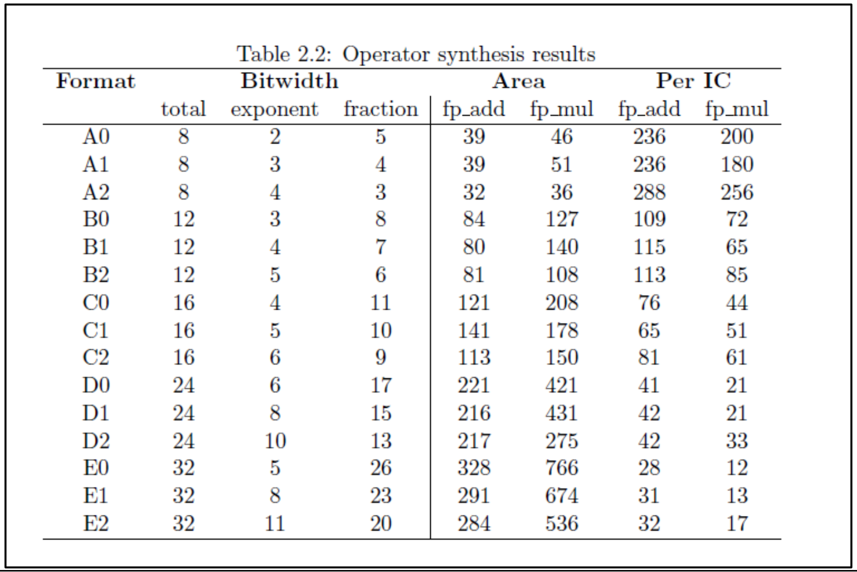


Table 2.2: Operator synthesis results
<table>
<tr><th>Format</th><th>Bitwidth</th><th></th><th></th><th>Area</th><th></th><th>Per IC</th><th></th></tr>
<tr><th></th><th>total</th><th>exponent</th><th>fraction</th><th>fp_add</th><th>fp_mul</th><th>fp_add</th><th>fp_mul</th></tr>
<tr><td>A0</td><td>8</td><td>2</td><td>5</td><td>39</td><td>46</td><td>236</td><td>200</td></tr>
<tr><td>A1</td><td>8</td><td>3</td><td>4</td><td>39</td><td>51</td><td>236</td><td>180</td></tr>
<tr><td>A2</td><td>8</td><td>4</td><td>3</td><td>32</td><td>36</td><td>288</td><td>256</td></tr>
<tr><td>B0</td><td>12</td><td>3</td><td>8</td><td>84</td><td>127</td><td>109</td><td>72</td></tr>
<tr><td>B1</td><td>12</td><td>4</td><td>7</td><td>80</td><td>140</td><td>115</td><td>65</td></tr>
<tr><td>B2</td><td>12</td><td>5</td><td>6</td><td>81</td><td>108</td><td>113</td><td>85</td></tr>
<tr><td>C0</td><td>16</td><td>4</td><td>11</td><td>121</td><td>208</td><td>76</td><td>44</td></tr>
<tr><td>C1</td><td>16</td><td>5</td><td>10</td><td>141</td><td>178</td><td>65</td><td>51</td></tr>
<tr><td>C2</td><td>16</td><td>6</td><td>9</td><td>113</td><td>150</td><td>81</td><td>61</td></tr>
<tr><td>D0</td><td>24</td><td>6</td><td>17</td><td>221</td><td>421</td><td>41</td><td>21</td></tr>
<tr><td>D1</td><td>24</td><td>8</td><td>15</td><td>216</td><td>431</td><td>42</td><td>21</td></tr>
<tr><td>D2</td><td>24</td><td>10</td><td>13</td><td>217</td><td>275</td><td>42</td><td>33</td></tr>
<tr><td>E0</td><td>32</td><td>5</td><td>26</td><td>328</td><td>766</td><td>28</td><td>12</td></tr>
<tr><td>E1</td><td>32</td><td>8</td><td>23</td><td>291</td><td>674</td><td>31</td><td>13</td></tr>
<tr><td>E2</td><td>32</td><td>11</td><td>20</td><td>284</td><td>536</td><td>32</td><td>17</td></tr>
</table>
</td></tr>
</table>

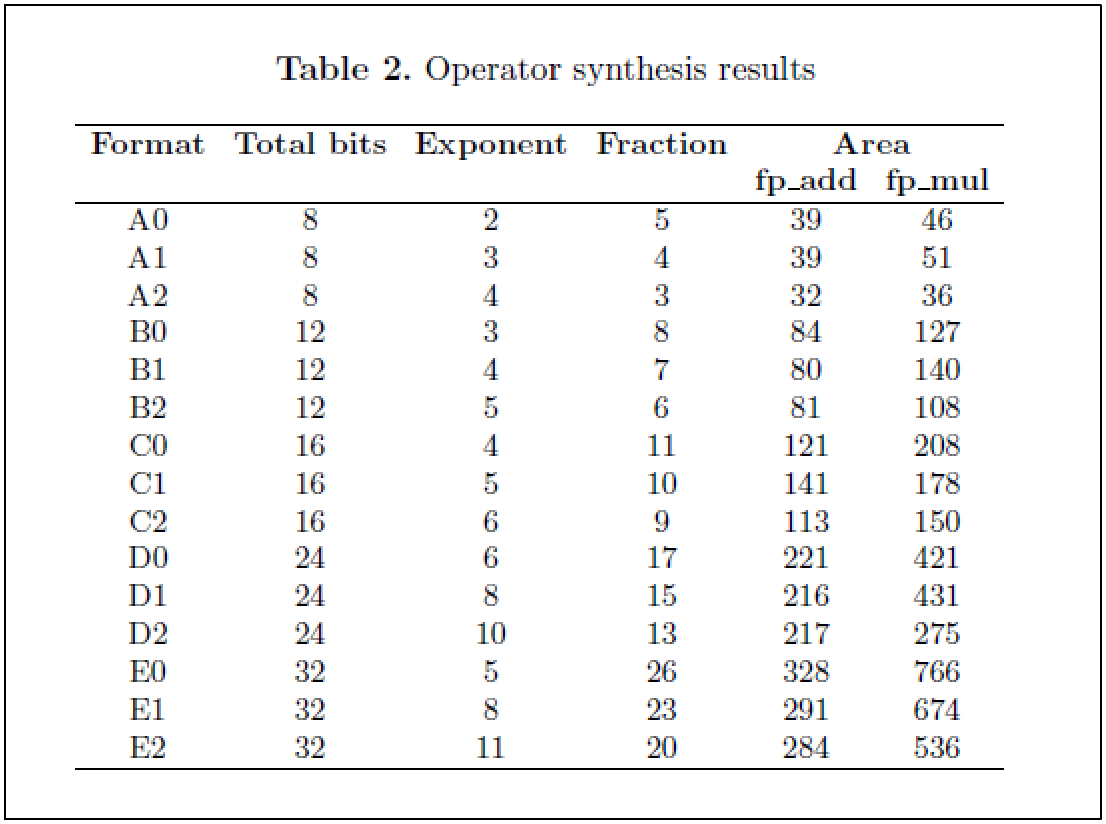
Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 7)	Exemplary Disclosure
	"The hardware modules described in Chapter 2 lend themselves to the creation of finely customized hardware implementations of algorithms. They give the designer full freedom to implement various sections of the algorithm in the most suitable arithmetic representation, be it fixed or floating-point. Also, bitwidths of all the signals in the circuit, whether in fixed or floating point representation, can be optimized to the precision required by the values the signal carries. When using floating-point arithmetic, the designer using the library has full control to trade off between range and precision. Because all the modules in the library are fully parameterized, the boundary between the exponent and fraction fields for the same total bitwidth is flexible. With a wider exponent field, the designer provides larger range to the signal, while sacrificing precision. Similarly, to increase the precision of a signal at the cost of reduced range, the designer chooses a narrower exponent and wider fraction field." Belanovic, *Library of Parameterized Hardware* at 50. Thus, the floating point library was capable of implementing various formats for any given bitwidth, and was not limited to the fraction and exponent combinations set forth in Table 2.2 of the Belanovic thesis. The formats implemented by the library could have included, for example, the floating point format, including the exponent and fraction widths described in the following: Sudha et al., *An Efficient Digital Architecture for Principal Component Neural Network and its FPGA Implementation* (2007) at 428 (each parameter in the design is represented as a floating point number with an 8-bit mantissa and an 8-bit exponent); Tong et al., *Reducing Power by Optimizing the Necessary Precision/Range of Floating-Point Arithmetic* (2000) at 273 (each parameter in the design is represented as a floating point number with a 5-bit fraction and 6-bit exponent); Texas Instruments TMS320C32 DSP (1995) (each parameter in the design is represented as a floating point number with a 7-bit fraction and 8-bit exponent); and Cray T3D System (1994) (each parameter in the design is represented as a floating point number with a 5-bit fraction and 11-bit exponent). "This line of thought was expanded on by the significant work of Shirazi et al. [17] who suggested application-specific formats for image and DSP algorithms in widths of 16

Exhibit 10 – Belanovic and Leeser

<table>
<tr><th>Claim Limitation (Claim 7)</th><th>Exemplary Disclosure</th></tr>
<tr><td></td><td>
(1-6-9) and 18 (1-7-10) bits, as opposed to the full 32 (1-8-23) bits in the IEEE standard.” Belanovic, <i>Library of Parameterized Hardware</i> at 18.

“Results for the fp add module in Table 2 also represent the fp sub module, which has the same amount of logic. The results in Table 2 show growth in area with increasing total bitwidth, for both modules.” Belanovic and Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i> at 7.


<b>Table 2.</b> Operator synthesis results
<table>
<tr><th>Format</th><th>Total bits</th><th>Exponent</th><th>Fraction</th><th colspan="2">Area</th></tr>
<tr><th></th><th></th><th></th><th></th><th>fp_add</th><th>fp_mul</th></tr>
<tr><td>A0</td><td>8</td><td>2</td><td>5</td><td>39</td><td>46</td></tr>
<tr><td>A1</td><td>8</td><td>3</td><td>4</td><td>39</td><td>51</td></tr>
<tr><td>A2</td><td>8</td><td>4</td><td>3</td><td>32</td><td>36</td></tr>
<tr><td>B0</td><td>12</td><td>3</td><td>8</td><td>84</td><td>127</td></tr>
<tr><td>B1</td><td>12</td><td>4</td><td>7</td><td>80</td><td>140</td></tr>
<tr><td>B2</td><td>12</td><td>5</td><td>6</td><td>81</td><td>108</td></tr>
<tr><td>C0</td><td>16</td><td>4</td><td>11</td><td>121</td><td>208</td></tr>
<tr><td>C1</td><td>16</td><td>5</td><td>10</td><td>141</td><td>178</td></tr>
<tr><td>C2</td><td>16</td><td>6</td><td>9</td><td>113</td><td>150</td></tr>
<tr><td>D0</td><td>24</td><td>6</td><td>17</td><td>221</td><td>421</td></tr>
<tr><td>D1</td><td>24</td><td>8</td><td>15</td><td>216</td><td>431</td></tr>
<tr><td>D2</td><td>24</td><td>10</td><td>13</td><td>217</td><td>275</td></tr>
<tr><td>E0</td><td>32</td><td>5</td><td>26</td><td>328</td><td>766</td></tr>
<tr><td>E1</td><td>32</td><td>8</td><td>23</td><td>291</td><td>674</td></tr>
<tr><td>E2</td><td>32</td><td>11</td><td>20</td><td>284</td><td>536</td></tr>
</table>

“The algorithm has been partitioned so the distance calculation is performed in floating-
</td></tr>
</table>

Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 7)	Exemplary Disclosure
	point arithmetic with 5 exponent and 6 fraction bits (1-5-6 format), while the comparison and accumulation operations are performed in 12-bit unsigned fixed-point format. Input data is in 12-bit unsigned fixed-point format, so it needs to be converted to the 1-5-6 floating-point representation." Belanovic and Leeser, *Library of Parameterized Floating-Point Modules and Their Use* at 8-9. "Also, all bitwidths in the datapath, whether in fixed or floating-point representation, can be optimized to the precision required for that signal. Hence, designing with our library of parameterized modules avoids expensive inefficiencies that are inherent in designs that operate only in the IEEE standard formats. In fact, such inefficiencies occur in any design that is restricted to using a small set of particular formats, even if these are custom. Using our library of parameterized modules provides the finest-grain control possible over datapath bitwidths. Finally, when using floating-point arithmetic, the designer has full control to trade off between range and precision. Because our modules are fully parameterized, the boundary between the exponent and the fraction for the same total bitwidth is flexible. Thus, with a wider exponent field, the designer has larger range for a value while sacrificing precision. Similarly, to increase the precision of a signal at the cost of reduced range, the designer chooses a narrower exponent and wider fraction field." Belanovic and Leeser, *Library of Parameterized Floating-Point Modules and Their Use* at 8-9. To the extent that Singular contends that this system does not itself identify a floating-point format that meets the particular range and error requirements, notwithstanding its disclosure of a floating point format with 9 fraction bits and 6 exponent bits, that format would have been obvious to one of skill in the art for the reasons explained in the Responsive Contentions Regarding Non-Infringement and Invalidity, which explain how those skilled in the art would mix and match formats depending on application specific needs. *See also* Belanovic, *Library of Parameterized Hardware* at 16 (explaining how custom datapaths for fixed- and floating-point arithmetic would have "optimal signal bitwidths throughout the custom datapath [that] are application-specific and depend on the values they carry.") For example, one of skill in the art would have understood the different combinations of fraction and exponent bits (*e.g.*, 5 fraction bits,

Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 7)	Exemplary Disclosure
	6 exponent bits, and one sign bit, for a total of 12 bits) would have been possible and even desired depending on the application. Alternatively, one of skill in the art would have been motivated to apply the teachings of Tong, which included a 5-bit mantissa and 6-bit exponent (*see* Tong chart) because Tong is cited. *See* Belanovic, *Library of Parameterized Hardware* at 73, n.21. *See also* Appendix to Responsive Contentions Regarding Non-Infringement and Invalidity (detailing error rates associated with different mantissa sizes).
[**156d**] at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit;	As reflected in Pavle Belanovic's thesis, *Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application*, and his related article with Miriam Leeser, *Library of Parameterized Floating-Point Modules and Their Use*, they made and used a system with at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit. *See, e.g.*: "**The Wildstar Reconfigurable Computing Engine** Reconfigurable computing is characterized by use of hardware elements that have reconfigurable architectures, as opposed to general purpose computing which uses hardware elements with fixed architectures. Many reconfigurable computing systems are based on one or more FPGAs connected to a number of memory banks. All designs presented in this project are implemented on the Wildstar reconfigurable computing engine from Annapolis Micro Systems. Figure 1.2 shows the structure of this board. Some of the main features of the Wildstar board are: • 3 Xilinx VIRTEX XCV1000 FPGAs, • total of 3 million system gates, • 40 Mbytes of SRAM, • 1.6 Gbytes/sec I/O bandwidth, • 6.4 Gbytes/sec memory bandwidth,

Exhibit 10 – Belanovic and Leeser

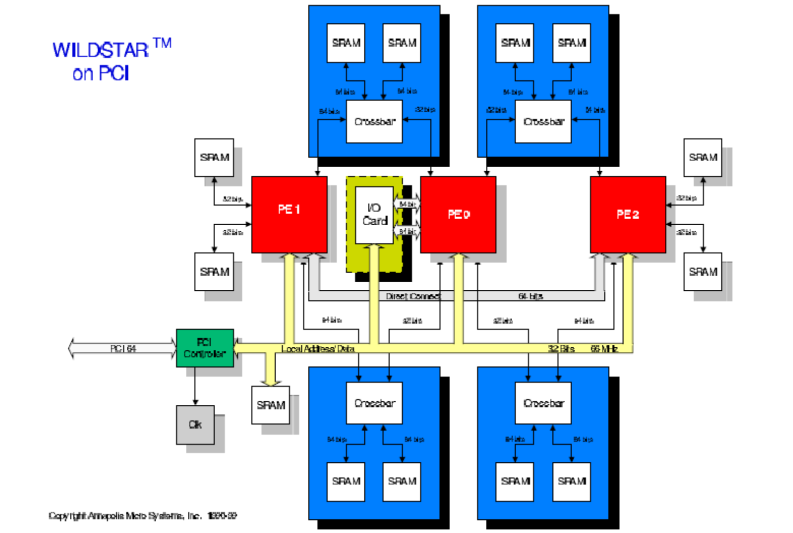
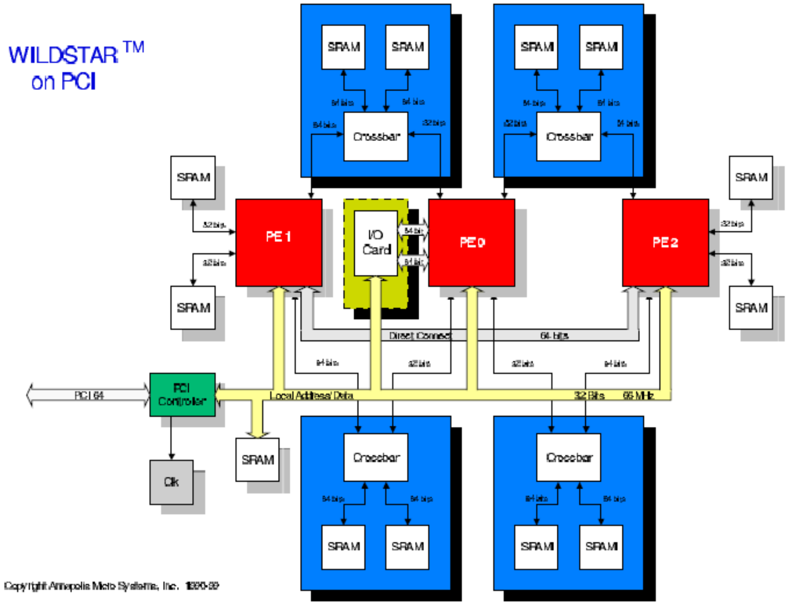
Claim Limitation (Claim 7)	Exemplary Disclosure
	- processing clock rates up to 100MHz.” Belanovic, *Library of Parameterized Hardware* at 14.  WILDSTAR TM on PCI SRAM Crossbar 64 bits 32 bits PE 1 PE 0 PE 2 I/O Card Direct Connect 64 bits PCI 64 PCI Controller Local Address/ Data 32 Bits 66 MHz Clk Copyright Annapolis Micro Systems, Inc. 1996-99 Figure 1.2: Structure of the Wildstar reconfigurable computing engine
[**156e**] wherein the at least one first computing device comprises at least one of a central processing unit (CPU), a graphics processing unit (GPU), a field programmable gate array (FPGA), a microcode-based processor, a hardware sequencer, and a state machine;	As reflected in Pavle Belanovic’s thesis, *Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application*, and his related article with Miriam Leeser, *Library of Parameterized Floating-Point Modules and Their Use*, they made and used a system with at least one first computing device comprises at least one of a central processing unit (CPU), a graphics processing unit (GPU), a field programmable gate array (FPGA), a microcode-based processor, a hardware sequencer, and a state machine. Specifically, Belanovic discloses a “host” computer that comprises at least a “state machine,” “FPGAs,” and various “processing units.” *See, e.g.,*: Belanovic, *Library of Parameterized Hardware* at 15 (Fig 1.2) (depicting the Wildstar

Exhibit 10 – Belanovic and Leeser

<table>
<tr><th>Claim Limitation (Claim 7)</th><th>Exemplary Disclosure</th></tr>
<tr><td></td><td>

computer engine, including (1) a PCI 64, which implies an interconnection with a PC and thus a CPU, and (2) an I/O Card and Controller, which qualifies as a state machine).



Copyright Annapolis Micro Systems, Inc. 1998-99

Figure 1.2: Structure of the Wildstar reconfigurable computing engine

“Field Programmable Gate Arrays (FPGAs) are integrated circuits with a flexible architecture, such that their structure can be programmed by the designer. FPGAs are composed of an array of hardware resources called configurable logic blocks (CLBs). The designer creates the functionality of the overall circuit by configuring CLBs to perform appropriate logic functions. Hence, FPGAs are a form of reconfigurable hardware, combining flexibility similar to software with the speed of specialized hardware.” Belanovic, *Library of Parameterized Hardware* at 13.

</td></tr>
</table>

Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 7)	Exemplary Disclosure
	"Typically, the parts of the algorithm that are assigned to software are serial or procedural in nature, while the highly parallel, computational parts of the algorithm get implemented in hardware. Custom datapaths are created in reconfigurable hardware to achieve desired functionality. Communication with the general purpose processor is done through memory banks and/or register tables in reconfigurable hardware, both of which are accessible by the custom hardware and the general purpose processor." Belanovic, *Library of Parameterized Hardware* at 15-16.
[**156f**] and, wherein the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide.	As reflected in Pavle Belanovic's thesis, *Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application*, and his related article with Miriam Leeser, *Library of Parameterized Floating-Point Modules and Their Use*, they made and used a system in which the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide. *See, e.g.*: "Unlike in fixed-point, in floating-point arithmetic multiplication is a relatively straight-forward operation compared to addition. This is due to the sign-exponent-magnitude nature of the floating-point format. The sign of the product is the exclusive OR (XOR) of the operand signs. The exponent of the product is the sum of the operand exponents. The mantissa is the product of the operand mantissas. Note that the operations on all three fields of the floating-point format are independent and can be implemented in parallel. The structure of the floating-point multiplier is shown in Figure 2." Belanovic and Leeser, *Library of Parameterized Floating-Point Modules and Their Use* at 4-5.

Exhibit 10 – Belanovic and Leeser

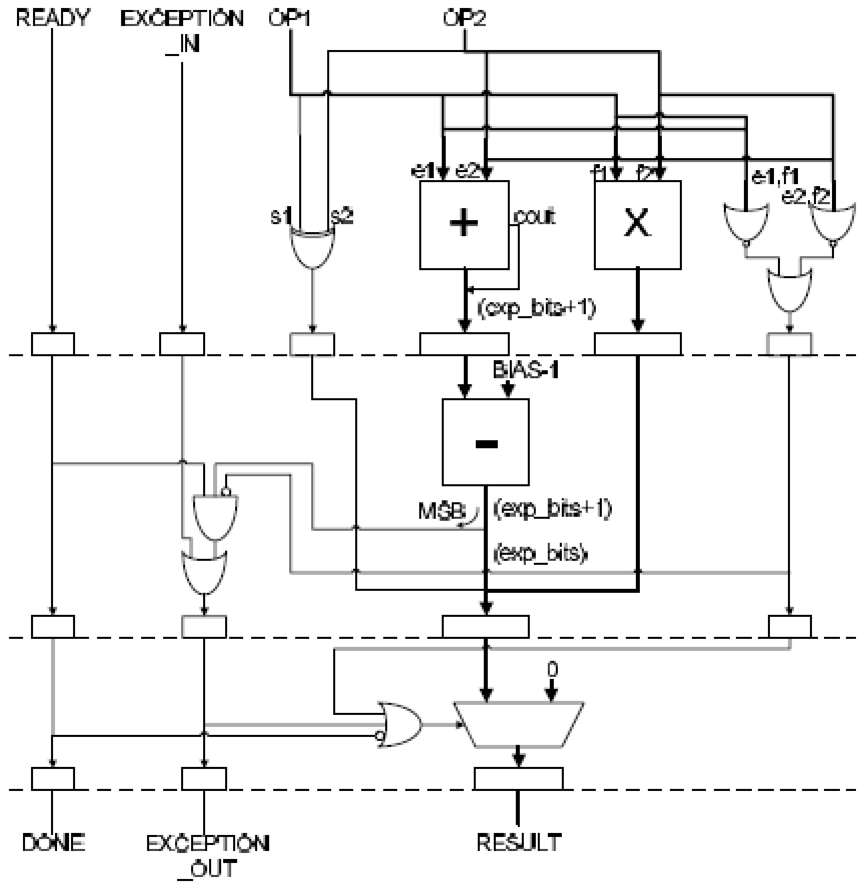
Claim Limitation (Claim 7)	Exemplary Disclosure
	 READY EXCEPTION _IN OP1 OP2 e1 e2 f1 f2 e1,f1 e2,f2 s1 s2 + cout X (exp_bits+1) BIAS-1 - MSB (exp_bits+1) (exp_bits) 0 DONE EXCEPTION _OUT RESULT Fig. 2. Floating-point multiplication "Synthesis results indicate that a realistic design on a Xilinx XCV1000 FPGA may include up to 31 addition or 13 multiplication operators, complete with denormalizing,

Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 7)	Exemplary Disclosure
	rounding and normalizing functionality each, for the IEEE single precision format. Similarly, a useful custom floating-point format, with 5 exponent and 6 mantissa bits for example, may provide the designer with up to 113 addition or 85 multiplication modules, all also complete with full format handling functionalities, on the same FPGA." Belanovic, *Library of Parameterized Hardware* at 34. "Some of the main features of the Wildstar board are: . . . 3 Xilinx VIRTEX XCV1000 FPGAs." Belanovic, *Library of Parameterized Hardware* at 14. "All these modules are specified in VHDL and implemented on the Wildstar reconfigurable computing engine from Annapolis Microsystems, using a Xilinx XCV1000 FPGA. Synthesis results for parameterized arithmetic operator modulesare presented in Table 2, for a set of floating-point formats labeled A0 through E2." Belanovic and Leeser, *Library of Parameterized Floating-Point Modules and Their Use* at 7. Thus, the system described by Belanovic and Leeser could have a total of 255 LPHDR execution units (85 multiplication modules per each of three FPGAs). "[I]t can be concluded that the three fields of the floating-point format do not interact during multiplication and can thus be processed at the same time, in parallel. The sign of the product is given as the exclusive OR (XOR) of the input value signs. Mantissa of the product is calculated by fixed-point multiplication of the input value mantissas, while the exponents of the input values are added to give the exponent of the product. To the extent that Singular contends that Belanovic's thesis does not identify a device with at least 100 multiplication execution units, notwithstanding its disclosure of a system with 255 such units, such a device would have been obvious given intervening FPGA developments for the reasons explained in the Responsive Contentions Regarding Non-Infringement and Invalidity.

Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 7)	Exemplary Disclosure
	"Often, much smaller bitwidths than those specified in the 754 standard are sufficient to provide the desired precision. Reduced bitwidth implementations require fewer resources and thus allow for more parallel implementations than using the full IEEE standard format. In custom hardware designs, it is possible, and indeed desirable, to have full control and flexibility over the exact floating-point format implemented." Belanovic and Leeser, *Library of Parameterized Floating-Point Modules and Their Use* at 2. "Each component is parameterized by exponent and mantissa bitwidths. Each component has a ready and a done signal to allow them to be easily assembled into larger designs." Belanovic and Leeser, *Library of Parameterized Floating-Point Modules and Their Use* at 3.

Exhibit 10 – Belanovic and Leeser

’273 Patent

Claim Limitation (Claim 53)	Exemplary Disclosure
[**273a**] A device:	As reflected in Pavle Belanovic’s thesis, *Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application*, and his related article with Miriam Leeser, *Library of Parameterized Floating-Point Modules and Their Use*, they made and used a system that consists of a device. *See* [**156a**]
[**273b**] comprising at least one first low precision high-dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,	As reflected in Pavle Belanovic’s thesis, *Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application*, and his related article with Miriam Leeser, *Library of Parameterized Floating-Point Modules and Their Use*, they made and used a system with at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first input signal representing a first numerical value to produce a first output signal representing a second numerical value. *See* [**156b**].
[**273c**] wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X % of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values	As reflected in Pavle Belanovic’s thesis, *Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application*, and his related article with Miriam Leeser, *Library of Parameterized Floating-Point Modules and Their Use*, they made and used a system with the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input. *See* [**156c**]. To the extent that Singular contends that this system does not itself identify a floating-point format that meets the particular range and error requirements, notwithstanding its disclosure of a floating point format with 9 fraction bits and 6 exponent bits, that format would have been obvious to one of skill in the art for the

Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 53)	Exemplary Disclosure
of that same input;	reasons explained in the Responsive Contentions Regarding Non-Infringement and Invalidity.
[**273d**] wherein the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide.	As reflected in Pavle Belanovic's thesis, *Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application*, and his related article with Miriam Leeser, *Library of Parameterized Floating-Point Modules and Their Use*, they made and used a system in which the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide. *See* [**156f**].

Exhibit 10 – Belanovic and Leeser

’961 Patent

Claim Limitation (Claim 4)	Exemplary Disclosure
[**961a**] A device comprising:	As reflected in Pavle Belanovic’s thesis, *Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application*, and his related article with Miriam Leeser, *Library of Parameterized Floating-Point Modules and Their Use*, they made and used a system that consists of a device. *See* [**156a**].
[**961b**] at least one first low precision high-dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,	As reflected in Pavle Belanovic’s thesis, *Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application*, and his related article with Miriam Leeser, *Library of Parameterized Floating-Point Modules and Their Use*, they made and used a system with at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first input signal representing a first numerical value to produce a first output signal representing a second numerical value. *See* [**156b**].
[**961c**] wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=10% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.2% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input; and	As reflected in Pavle Belanovic’s thesis, *Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application*, and his related article with Miriam Leeser, *Library of Parameterized Floating-Point Modules and Their Use*, they made and used a system with the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input. *See* [**156c**]. To the extent that Singular contends that this system does not itself identify a floating-point format that meets the particular range and error requirements, notwithstanding its disclosure of a floating point format with 9 fraction bits and

Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 4)	Exemplary Disclosure
	6 exponent bits, that format would have been obvious to one of skill in the art for the reasons explained in the Responsive Contentions Regarding Non-Infringement and Invalidity.
[**961d**] at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit.	As reflected in Pavle Belanovic's thesis, *Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application*, and his related article with Miriam Leeser, *Library of Parameterized Floating-Point Modules and Their Use*, they made and used a system with at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit. *See* [**156d**].

Claim Limitation (Claim 13)	Exemplary Disclosure
[**961e**] A device comprising:	As reflected in Pavle Belanovic's thesis, *Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application*, and his related article with Miriam Leeser, *Library of Parameterized Floating-Point Modules and Their Use*, they made and used a system that consists of a device. *See* [**156a**].
[**961f**] a plurality of components comprising:	As reflected in Pavle Belanovic's thesis, *Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application*, and his related article with Miriam Leeser, *Library of Parameterized Floating-Point Modules and Their Use*, they made and used a system with at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit. *See above* [**156d**].
[**961g**] at least one first low precision high-dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical	As reflected in Pavle Belanovic's thesis, *Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application*, and his related article with Miriam Leeser, *Library of Parameterized Floating-Point Modules and Their Use*, they made and used a system with at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first input signal representing a first numerical value to produce a first output

Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 13)	Exemplary Disclosure
value,	signal representing a second numerical value. *See* [**156b**].
[**961h**] wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=10% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.2% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input.	As reflected in Pavle Belanovic's thesis, *Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application*, and his related article with Miriam Leeser, *Library of Parameterized Floating-Point Modules and Their Use*, they made and used a system with the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input. *See* [**156c**]. To the extent that Singular contends that this system does not itself identify a floating-point format that meets the particular range and error requirements, notwithstanding its disclosure of a floating point format with 9 fraction bits and 6 exponent bits, that format would have been obvious to one of skill in the art for the reasons explained in the Responsive Contentions Regarding Non-Infringement and Invalidity.